

TARTU ÜLIKOOL
Loodus- ja tehnoloogiateaduskond
Füüsika instituut

Hannes Tamme

**Protsessi juhtimise tarkvaraline raamistik
puidu konvektiivkuivatuse ja spektroeletrokeemia näidetel**

Magistritöö

Juhendajad: Madis Noppel, füs.-mat. kandidaat
Valdek Tamme, PHD student

Sisukord

1	Sissejuhatus	3
2	Eesmärk	3
3	Arhitektuur.....	4
4	Juhtimissüsteemi komponendid.....	4
4.1	Sündmuste korrellaator	5
4.1.1	Sündmuste korrellaator – detailid	5
4.2	Tarkvara manipulaator	6
4.2.1	GUI tarkvara manipulaator – detailid	7
4.2.2	CLI tarkvara manipulaator – detailid.....	7
4.3	Haldusliides.....	8
4.3.1	Haldusliides – detailid.....	9
4.4	Konfiguratsioonihaldus.....	10
4.4.1	Konfiguratsioonihaldus – detailid.....	10
4.5	Tugitarkvara.....	11
4.5.1	Abiskriptid	11
4.5.2	Võrguliides.....	12
4.5.3	Raudvaramoodul	13
4.5.4	Virtualiseerimine.....	13
4.5.4.1	Miimumnõuded virtualiseeritud masinale	15
5	Näited.....	16
5.1	Takistusmõõtja ”Scantronik Gigamodule”, andmelogi ”Scantronik Termofox” ja ”RasperryPi” GPIO ümberlüüti liidestamine	16
5.2	Kaalu ”Precisa” ja miniarvuti ”RasperryPi” liidestamine.	17
5.3	“AutoLab PGSTAT 30” kui elektrodide potentsiaali kontrolleri, “Spectrum GX” infrapuna spektromeeter’i liidestamine.....	17
6	Kokkuvõte	20
7	Summary.....	21
8	Viited	23

1 Sissejuhatus

Keerukamate ja pikema ajalise kestvusega eksperimentide läbi viimisel kaasaegses teaduslaboris (või ka välimõõtmistel), tekib sageli probleeme kasutatavate mõõteseadmete raudvara ja tarkvara ühendamisel ühtseks uurimistöö eesmärgi täitvaks tervikuks. Teiseks, sageli tekib vajadus eksperimendi käigus mõni seade ümber seadistada järgmisele mõõtereziimile, siis järgmisele, jne., mis omakorda tähendab enamasti nii tarkvaralist kui ka raudvaralist seadistamist. Probleeme esineb ka keerukamate eksperimentide puhul andmevoogude haldamisega. Selle asemel, et kõik andmed jookseksid sünkroonselt ühte ühisesse andmetabelisse (baasi), satuvad andmed tihti mitmesse erinevasse arvutisse (mis võivad omakorda olla erinevast põlvkonnast või erineva op-süsteemiga), erinevatesse kataloogidesse, jne. Käsitsimõõtmistena teostatud keerukamate eksperimentide korral, mille ühiseks tunnuseks on mitmesuguste seadmetega lähedastel ajahetkedel järjestikuselt või paralleelselt sünkroonselt mõõtmine (vt. Romann et al.,2013; Tamme et al.,2010,2011,2012), oleks otstarbekas teaduslabori ressursside parema fokuseerimise huvides teadusuuringu protsessi teataval määral automatiseerida, säilitades siiski kõigi eksperimendis kasutatavate seadmete tehnilistes spetsifikatsioonides ette nähtud raudvaralise ja tarkvaralise individuaalsuse.

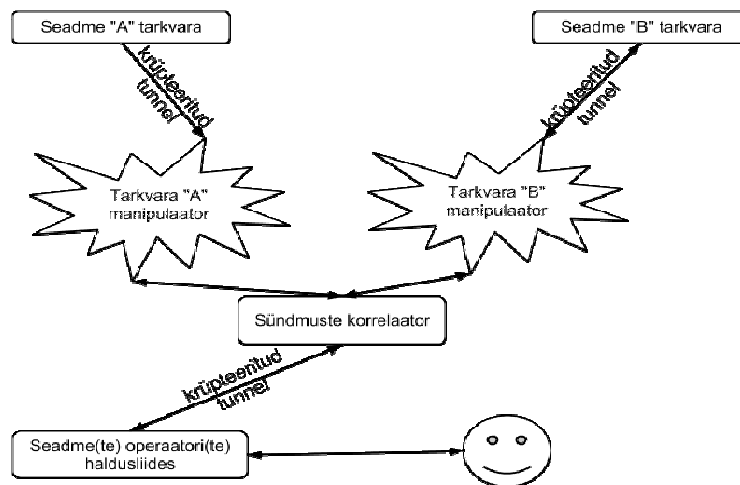
2 Eesmärk

Teaduslaboritesse koguneb aja jooksul seadmeid, mis on toodetud eri aegadel erinevate firmade poolt, millede sisend ja väljundliides arvutiga on seetõttu paratamatult väga erinevas formaadis. See puudutab ka aja jooksul omandatud tarkvara. See asjaolu muudab nende seadmete ja tarkvara lülitamise ühtsesse süsteemi, mis on vajalik, et teostada kompleksseid mõõtmisi, kus sünkroniseeritult mõõdetakse ajas vahelduva sagedusega väga erinevaid parameetreid, väga tülikaks. Antud töö eesmärgiks on, kasutades vabavaralisi tarkvaraplokke, panna kokku ja häälestada süsteem, mida kasutades saab eksperimentaator oma mõõteülesande kirja panna lihtsas keeles, mõtlemata ühilduvusprobleemidele. Realiseerida süsteem konkreetsetel seadmepargil - EMÜ Metsandus- ja maaehitusinstituudi (MI) metsatööstuse osakonna puiduteaduse labori ja Chemicum'i (TÜ keemiainstituut) labori seadmetel. Paljudel mainitud seadmetel puudub tarkvaratugi, mis võimaldaks kasutada seatud eesmärgiga analoogseid ülesandeid täitvaid kommertsiaalseid tarkvarapakette nagu Labview.

Süsteemi realiseerimine näitas, et süsteemi kompileerimine ja paigaldamine on üsna ajakulukas. Seetõttu on süsteem paigaldatud virtuaalmasinasse. See võimaldab vajadusel süsteemi kiiret taaspaigaldamist arvutisse kui ka kiiret süsteemi paigaldamist uutesse arvutitesse. Järgnevas on antud detailsem ülevaade realiseeritud süsteemist.

3 Arhitektuur

Alltoodud skeemil on kujutatud süsteemi komponentide omavahelised seosed.



Joonis 1. Süsteemi komponentide omavahelised seosed.

4 Juhtimissüsteemi komponendid

Järgnevates alampeatükkides on kirjeldatud süsteemis kasutatavaid tarkvarakomplekte ja nende omavahelisi seoseid. Alamjaotis “detailid” on mõeldud süsteemi seadistamisega tegelejatele ega ole vajalik süsteemi operaatoritele - eeldusel, et mõlemad ei ole sama isik. Alamjaotis “failid” sisaldavad seadistusfailide asukohta failisüsteemis ja konkreetse faili funktsionaalsuse kirjeldust. Enamikele skriptidele on kirjutatud ingliskeelne manual, mis on kätte saadavad kasutades programmi `man [man(1)]`. Või Operaatori haldusliidest sisestades `?programminimi`.

Kasutaja kuulub gruppi `sudo` [SUDO(8)] ja kasutajale lubatud liigutused on kirjeldatud failis `/etc/sudoers` [SUDOERS(5)]. Logimisel käivitatakse mõningaid ‘majapidamistööd’ juurkasutaja õigustes. Tööde nimekiri on leitav failist `/etc/rc.local`.

4.1 Sündmuste korrellaator

See komponent tõlgib erinevates formaatides andmevooge vastavalt tarkvara/operaatori sisendi väljundi ootustele. Sündmuse korrellaator „kuulab“ failide sisu muutuseid ja võrdleb neid etteantud regulaaravaldistega. Sündmuse (kirje tekkimisel faili) esinemisel (puudumisel) tehakse ettenähtud manipulatsioon vastava tarkvaraga. Kautatavad regulaaravaldised on enamasti perl'is kasutatava süntaksiga [PERL(1)].

4.1.1 Sündmuste korrellaator – detailid

Sündmuste korrellaatoriks on tarkvara nimega SEC [sec(1)]. Viimane on loodud Risto Vaarandi [RV] poolt logifailide analüüsiks. SEC'i pakutakse koos põhjaliku dokumentatsiooniga, millele pääseb ligi konsoolist sisestades “man sec”. Toodud tarkvarakomponent võimaldab meeles pidada ja reageerida erineva “mustriga” sündmustele, mille algmaterjal võetakse failidest mis programmi käivitamisel initseerimise skriptis ette antakse. Lisaks erinevate mõõteseadmete sisendi väljundi korrektsuse kontrollile toimub selle abil ka masina (te) enda üldise ”tervise” kontroll. Taaskäivitatakse liigse ressursikasutusega programme, kontrollitakse välise maailmaga suhtlemiseks mõeldud kanali turvalisust. Vajadusel kirjutatakse välise võrguliidese tulemüüri reegleid ümber, saadetakse proleemsete kohtade kohta meile vastavasisulise raportiga, käivitatakse perioodilisi töid jne. Kuna SEC on vormitav 4-5 realiste käskudega, siis see teeb temast äärmiselt paindliku tööriista, mille kasutamist piirab ainult kaine mõistus. Konfiguratsioonifailides muudatuste tegemisel on vajalik programmi uuesti käivitamine või nn. soft-restart, kus konfiguratsioon loetakse uuesti sisse kui ei kustutata eelneva töö käigus mällu loetud muutujate hetkeväärtusi.

Failid:

`/etc/init.d/sec/sec_ui.conf`

Sündmuse korrellaatori põhiline konfiguratsioonifail. Siin on kirjas regulaaravaldised millele regeeritakse ja alamprogrammid mida vajadusel käivitatakse. Vastavalt süsteemi eripäradest tingitud vajadustele saab lisada ka teisi seadistusfaile. Siis tuleb teha vastav muudatus ka järgnevalt toodud käivitamis peatamisskripti.

`/etc/init.d/sec/sec.sh`

Sündmuse korrelaatori käivitamise, peatamise ja konfiguratsiooni laadimise skript. Ilma parameetriteta skripti käivitamine annab valiku võimalikest valikutest ja ka lühida kirjelduse valiku tulemist. Võimaldab peatada käivitada SEC'i ja programmi sisemiste muutujate hetkeväärtust kirjutada faili. Viimane valik on kasulik konfiguratsioonifailide vigade otsimiseks, kui programm on käivitatud taustal [DAEMON(3)].

/home/manager/.config/autostart/sec.sh.desktop

GNOME [gnome] aknahalduri käivitamisel automaatselt starditakse ka SEC [SEC(1)] kasutaja manager õigustes. Sellistest õigustest enamasti piisab sest vaikimisi umask [UMASK(2)] on 022.

4.2 Tarkvara manipulaator

Tekitab kindlaksmääratud formaadiga andmevoo tarkvaralt tarkvarale. Kuna tegemist on valdavalt kinnise lähtekoodiga tarkvaraga mida on vaja automaatselt käsitleda, siis ei saa legaalselt programmi sisse vaadata ja läbi tuleb ajada tootja poolt pakutavate vahenditega. Juhtimissüsteem sisaldab tööriistu, millega saab manipuleerida graafilist kasutajaliidest, anda käsurearakenduele interaktiivse (expect [EXPECT(1)]) sessiooni käigus arusaadavat sisendit. Tarkvara manipulaatori komponent on kogu süsteemi kõige keerulisem osa. Seda selle tõttu, et iga manipulaator on enamasti unikaalne ja selle tõttu ei ole kuskilt võtta soovitusi ega lahendusnäidiseid.

Tarkvara manipulaator koosneb enamasti mitmetest alamprogrammidest, skriptidest mis täidavad oma kindlat ülesannet alamülesannet. Lähtutud on filosoofiast, et üks programm teeb ühte kindlat asja. Ülesanne loetakse sooritatuks kui õnnestub SEC'i poolt kuulatavasse faili tekitada infot kandev tähejada. Ühe keskmise keerukusega ülesande illustreerimiseks võib tuua järgnevate liigutuste jada.

Graafilise kasutajaliidese soovitud ekraaniregioon kopeeritakse pildifaili. Viimaselt loetakse välja tekst. Saadud tulem antakse edasi raudvaraga suhtlevale mikrokontrollerile. Viimane teeb soovitud liigutuse raudvaraga ja tagastab tulemi. See omakorda suunatakse faili, mida kuulatakse sündmuse korrelaatoriga. Tulem töödeldakse ja antakse edasi soovitud programmile sisendiks.

Protsessi jälgimiseks on kasutusel tagurpidi ssh [SSH(1)] tunnel mõnda välisvõrgust kätte saadavasse masinasse (sageli math.ut.ee). Viimases piisab ühenduse tegemisest eelnevalt valitud porti, et pääseda soovitud ressursile ligi.

4.2.1 GUI tarkvara manipulaator – detailid

Graafilise keskkonnaga manipuleerimiseks kasutatakse tarkvara nimega Sikuli [SIKULI(1)]. Viimane on OpenCV [x] funktsionaalsust kasutav pildituvastusprogramm, mille kontrollskripte kirjutatakse python'is. Sikuli omab kasutajaliidest milles on võimalik lihtsamaid programme valmistada kasutades ettevalmistatud ehitusblokke. Kui vajaminevad operatsioonid on keerukamad kui Sikuli poolt pakutav arenduskeskkond võimaldab tekitada, siis saab ka käsitsi koodi lisada. Koodi valideerimiseks on haldusliidesesse lisatud vastav funktsionaalsus pyflakes [PYFLAKES(1)] näol. Pildi pealt teksti ekstraktimiseks kasutatakse OCR tesseract [TESSERACT(1)] mootorit. Eelnevad pildimanipulatsioonid teostatakse ImageMagic [ImageMagick(1)] poolt pakutava tarkvaraga.

Pildil olevate objektidele (oksakohad, lõhed) tekstina esitamiseks on kirjutatud QVPO [ts(7)] nimeline tarkvara mis võimaldab peale kalibreerimist läbi graafilise kasutajaliidese lisada objektide parameetrid (mõõtmed, asukoht, kogus).

Failid:

`/usr/local/bin/sikuli/sbus.sikuli/sbus.py`

Graafilise kasutajaliidese automatiseerimise skript.

`/usr/local/bin/sikuli/krdc.sikuli/krdc.py`

RDP / VNC ühenduste graafilist kasutajaliidest automatiseeriv skript. Automatiseeritavaks kasutajaliideseks on valitud krdc [krdc] . Kasutajaliidest-mitte käsurida kasutatakse seetõttu, et käsurida ei võimaldanud defineerida mitut masinat, millesse ühendus tehakse.

4.2.2 CLI tarkvara manipulaator – detailid

Käsurearakenduse automatiseerimine on realiseeritud erinevate Debiani [DEBIAN] pakihaldussüsteemis olevate rakendustega. Interaktiivsete sessioonide korral on kasutatav expect [EXPECT(1)]. Enamasti suunatakse läbi fifo [FIFO(7)] ühe programmi väljund teise sisendisse.

Vajadusel liigub info ka läbi sündmuse korrelaatori. Kasutatavaid pakkide nimekirja hoitakse samas versioneerimissüsteemis kus ülejäänud konfiguratsiooni.

Kõik käsurearakendused kasutavad logimiseks bash'i [BASH(1)] skripti (predate.sh) mis lisab igale reale ajamargi ja programmi nime, mille poolt info logifaili tuli. Eeltoodud skript käivitatakse sündmuse korrelaatori poolt. See ajamark on vajalik hilisemaks andmete sorteerimiseks programmi, kuupäeva järgi. Kasutatakse nn. mõõteinfo koondtabelite tekitamiseks. Ehk siis kui on võimalik ühte andmetabelisse koondada erinevate sensorite, seadmete poolt toodetud infot. Ja seda näiteks GNU R [R(1)] statistikapaketiga edasi töödelda või csv'na [RFC 4180] soovitud kohta edasi eksportida.

Failid:

`/usr/local/bin/expect.sh`

Skript mis võimaldab automatiseerida interaktiivset käsurearakendust.

`/usr/local/bin/grep_r/*`

Skriptid mis tekitavad erinevaid väljundformaate. Lisainfot nende skriptide toimimise kohta tasub otsida järgnevast dokumentatsioonist: [GREP(1)],[SED(1)],[MAWK(1)],[TR(1)],[CUT(1)].

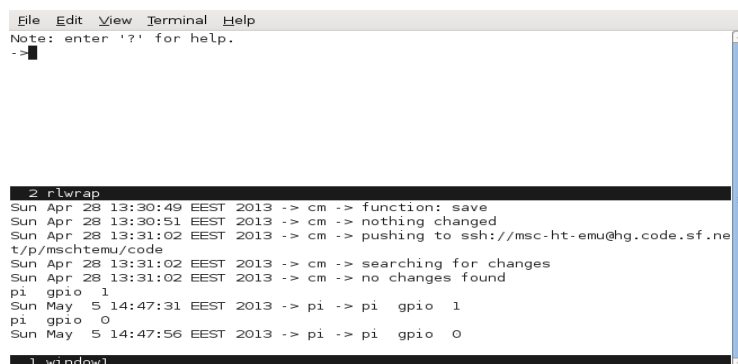
4.3 Haldusliides

Operaatori käsutuses olevad tekstifailid, millele saab kas käsitsi või mõne muu programmiga kirjutada. Haldusliideseks võib olla suvaline tekstiredaktor, millega lisatakse faili soovitud käskusid. Eeltoodud viis on siiski reaajas üsna ebamugav teostada, sest samasse tekstifaili lisavad ka paljud alamprogrammid oma väljundit. Seetõttu on soovitatav kasutada süsteemiga kaasa pandud haldusliidest. Viimasesse on sisse ehitatud ka dokumentatsiooni sirvimine, viimaste käskude mälu ja reaajas faili sisu kuvamine. Haldusliides käivitatakse siis, kui masinasse sisse logitakse (konsooliga). Haldusliideses antud käsud suunatakse edasi vajadusel ekraan nullile kasutaja 'manager' õigustes (xauth [XAUTH(1)]). Haldamise teeb mugavamaks linuxi virtuaalmasina ja rauast masina info kuvamine kahel eraldi ekraanil.

Haldusliides on soovitatav käivitada mõnes teises ekraani omavas arvutis. Seda sellepärast, et muidu võib tekkida olukord, kus klaviatuuri ja hiirt kontrolliv tarkvara hakkab võistleva süsteemi operaatoriga. Kahe ekraani lahendus on kasutusel TÜ Keemia instituudis (Chemicum). Eesti Maaülikoolis (täpsemalt EMÜ Metsandus- ja maaehitusinstituudi (MI) metsatööstuse osakonna puiduteaduse laboris) on süsteemi juhtarvutil linux'i põhi ja seal on LXDE aknahalduris seadistada sobiv hulk virtuaalseid töölaudu. Käesoleval hetkel on kasutuses kolm tükki. Üks rauast masina jaoks ja ülejäänud virtualiseeritud süsteemide jaoks. Chemicumis tehakse putty[putty(1)] abil ssh ühendus juhtimissüsteemi virtuaalmasinasse. Ühendumisel manager kasutajaga võetakse vaikimisi lahti juhtimissüsteemi operaatori haldusliides. Vajalikud parameetrid selleks on kasutaja manager kodukataloogis asuvas .bashrc skriptis. Selline kasutusviis välistab kasutaja ja juhtimissüsteemi programmide sisend/väljund seadmete pärast võistlemise (vt. eeltoodut).

4.3.1 Haldusliides – detailid

Haldusliides (vt pilti 1) koosneb screen [SCREEN(1)] tarkvaras jooksvatest shelli skriptidest. Kasutaja sisendit loetakse vastava bash'i skriptiga faili, mida kuulab SEC väljastatakse programmiga tail [TAIL(1)]. Ehk siis ülemises aknapooles oodatakse operaatori sisendit ja alumises aknapooles kuvatakse konfiguratsioonifailidesse lisanduvaid ridu. Eelnevate käskudele ligipääsu eest hoolitseb rlwrap [rlwrap(1)]. Manuaale väljastatakse juhtimissüsteemi haldusliideses programmiga man [MAN(1)]. Graafilise ekraanipildi suunamiseks eksporditakse xauth [XAUTH(1)] "küpsised" haldusliidese käivitamisel ja suunatakse väljund ekraan nullile ('Screen0'). Haldusliidesse on sisse ehitatud abi saamise funktsionaalsus. Abi saab kui sisestada "?" või "?soovitud_käsk". Küsimärgi sisestamisel kuvatakse kõikide käskude süntaks.



```
File Edit View Terminal Help
Note: enter '?' for help.
->

2 rlwrap
Sun Apr 28 13:30:49 EEST 2013 -> cm -> function: save
Sun Apr 28 13:30:51 EEST 2013 -> cm -> nothing changed
Sun Apr 28 13:31:02 EEST 2013 -> cm -> pushing to ssh://msc-ht-emu@hg.code.sf.net/p/mschtemu/code
Sun Apr 28 13:31:02 EEST 2013 -> cm -> searching for changes
Sun Apr 28 13:31:02 EEST 2013 -> cm -> no changes found
pi gpio 1
Sun May 5 14:47:31 EEST 2013 -> pi -> pi gpio 1
pi gpio 0
Sun May 5 14:47:56 EEST 2013 -> pi -> pi gpio 0

1 window1
```

Pilt 1. Chemikumis ja Maaülikoolis kasutusel olev süsteemi haldusliides.

Failid:

/home/manager/.while_read.sh

Loeb operaatori sisendit ja edastab info sündmuse korrelaatorile. Mis omakorda käivitab vajalikud alamprotseduurid. Samuti võimaldab ligipääsu erinevate programmide manuaalidele ja annab esmast abiinfot.

/home/manager/.screenrc

Screen koopiaid käivitav skript. Loob haldusliidese toimimiseks vajaliku keskkonna. Jagab ekraani osadeks millest ülemisse poolde oodatakse operaatori sisendit ja alumisse poolde kuvatakse sündmuse korrelaatorile antud sisendit ja töödeldud väljundit.

4.4 Konfiguratsioonihaldus

Süsteemi seadistamiseks on mõistlik kasutada versioneerimistarkvara. Viimane võimaldab hoida järge baaskonfiguratsioonis pakutavate failide muutustes.

Kuna versioneerimistarkvara kasutamine ei ole teema, mida süsteemi operaator peaks valdama, siis on kasutajamugavuse huvides haldusliidesele lisatud soovitud operatsioone võimaldav funktsionaalsus. Konfiguratsioonihaldus on kasutatav läbi haldusliidese. Lisainfot saab kui kirjutada haldusliidsesse ‘?cm’.

4.4.1 Konfiguratsioonihaldus – detailid

Konfiguratsiooni all peetakse silmas paigaldatud pakkide nimekirja, seadistusfaile ja skripte. Eeltooduid hoitakse SourceForge [SourceForge] Mercuriali [HG(1)] repositooriumis. Konfiguratsioonihaldur [CM(8)] on skript, mis võimaldab operaatoril salvestada (‘cm save’) muudatusi ja liikuda konfiguratsiooni erinevate versioonide vahel (‘cm upgrade’ , ‘cm downgrade’). Silmas peab pidama, et konfiguratsioonis saab tagasi liikuda nii kaua kuni konfiguratsiooni halduse skript mercuriali repositooriumisse lisati ja ‘cm upgrade’ peetakse silmas süsteemi uuendamist kuni viimase versioonini, mis on kätte saadav.

Kummalisel kombel on osutunud väga keeruliseks süsteemi kasutajatele selgitada oma failides tehtud muudatuste salvestamise olulisust. Samas käesoleva töö autor ei taha tagaplaanil teha

pealesunnitud uuendusi ja salvestamisi, kuna see piiraks kasutaja valikuvabadust. Kasutajale on jäetud süsteemi üle täielik kontroll. Isegi siis kui otsused ei ole alati optimaalsed.

Failid:

`/usr/local/bin/cm`

Konfiguratsioonihalduse skript läbi mille on pikemat käsurida nõudvad ülesanded enamasti paari sõnaga sooritatavad läbi haldusliidese.

`/usr/local/share/man8/cm.8`

Konfiguratsioonihalduse manual. Kätte saadav läbi haldusliidese või otse konsoolist ('man cm').

`/var/local/packages`

Paigaldatud kõikide *.deb pakside nimekiri. Töö kirjutamise hetkel ei sisalda see käsitsi paigaldatud tarkvara. Sellele probleemile peab leidma mingi mõistliku lahenduse. Seda nimekirja uuendatakse konfiguratsioonihalduse skriptiga.

4.5 Tugitarkvara

Tugitarkvara all on käesoleval juhul silmas peetud programme, mis ei ole otseselt vajalikud süsteemi käivitamiseks, kuid ilma milleta ei ole funktsionaalsust, mis teeb süsteemi kasutatavaks.

4.5.1 Abiskriptid

Järgnevalt toodud skriptid/programmid on enamasti süsteemispetsiifilised. Ehk kui süsteemi kasutajal ei ole seonduvat tarkvara/raudvara, siis ei ole ka tarvilik paigaldada vastavat skripti. Samas kettapinna kokkuhoid ei ole märgatav mõne kB skripti paigaldamata jätmisel, seega on enamasti lisatud kõik, mis hetkel olemas on. Samuti võib erinevates süsteemides olev sama programm/skript olla erineva seadistusega. Täpne info on mercuriali repositooriumis sourceforge poolt pakutavas keskkonnas.

Failid:

`/usr/local/bin/predate.sh`

Lisab logifailile igale uulele kirjele ajamargi ja programmi nime kelle poolt info tuli.

`/usr/local/bin/image2text.sh`

Loeb pildi pealt teksti ja väljastab soovitud faili. Chemicumis tehakse sobivast ekraanimiirkonnast tõmmis, mis antakse läbi SEC'i peamiselt Imagemagic tööriistadest koosnevale eelredigeerimiskriptile ja seejärel vastavate parameetritega Tesseract OCR [OCR] mootorile. Tulem tagastatakse läbi SEC' poolt kuulatava faili.

/usr/local/bin/fft.sh

Kuna süsteemis liikuvast infost sisaldub ka ebasoovitatav mürasignaal siis on süsteemi lisatud erinevaid heuristilisi filtreid sisaldav moodul. Hetkel ei ole kuskil kasutusel. Skript kasutab GNU R [R(1)] poolt pakutavaid fft moodulit ja kompleksarvudega manipuleerimise võimalusi.

/usr/local/bin/ahlborn.sh

Ahlborn 9 kanaliga andmelogeri manipuleerimise skript. Seade väljastab andmeid jadaporti (rs232).

/usr/local/bin/qvpo

Pildil olevate objektide loendur klassifitseerija. See programm võimaldab piltidel olevatele objektidele anda iseloomustavad parameetrid.

/usr/local/bin/grep_r/grep.sh

Saadab STDOUT'i ja soovi korral logifaili vastava programmi poolt soovitud ajavahemikus tekitatud kirjed. Kuna mitmenädalaste failide maht on suur, siis on keskne roll täita GNU parallel [parallel] paketil. See eeldab andmete kätte saadavaks tegemist raudvaralisele linuxile, et mitmest tuumast kasu lõigata.

/usr/local/bin/precisa.sh

See skript loeb sisse kaalu Precisa poolt jadaporti (rs232) tulevaid andmeid ja edastab info logifaili, mida kuulab sündmuse korrelaator SEC.

4.5.2 Võrguliides

Kuna juhtimissüsteemis olevate objektide füüsiline asukoht ei pruugi olla samas ruumis, siis on käesolevas töös kasutatud vahendeid, mis toimivad üle võrgu. See lahendus eeldab, et välja paistev süsteemi ots, mida kasutatakse vajadusel haldamiseks, on mingil mõistlikul viisil piiratud ligipääsuga. Süsteemi sisenemine toimub usb mälupulkade peal jagatud krüprovõtmetega, mille avalik osa asub failis ~/.ssh/authorised_keys. Võtmete laadimine võtmerõngasse toimub

programmi ssh-add [SSH-ADD(1)]. Ruuteri/tulemüüri/dhcp/dns serveri osa täidab miniarvuti RaspberryPi [RaspberryPI]. Liidesle liikuvate pakettide filtreerimine on teostatud kasutades iptables [IPTABLES(8)].

Failid (RaspberryPi):

/home/pi/.ssh/*

Siin on erinevatesse kohtadesse minevaid krüptovõtmeid [SSH-KEYGEN(1)].

/home/pi/wireless/*

Siin on skriptid mis võimaldavad laadida kernelisse wifi mooduleid, kui kohalik võrgupoliitika ei võimalda kasutada kaablit.

4.5.3 Raudvaramoodul

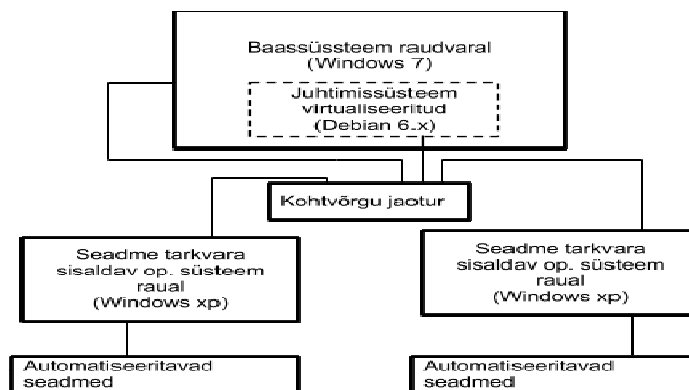
Raudvaraga (türistorreleed, pinge/vooluväljunduga sensorid jms.) suhtlemiseks on kasutusel Parallax poolt toodetud p8x32A [p8x32a] mikrokontroller ja GPIO'd [GPIO] omav miniarvuti RaspberryPi [RaspberryPI]. Kahele eeltoodud plaadile lisatakse tarvidusel vajalike komponente. Raudvara juhtimine käib üle ssh tunneli, läbi RaspberryPi. Vajaminev infot liigutatakse läbi SEC'i poolt kuulatavate logifailide ja töödeldakse vastavalt.

Töö kirjutamise hetkel on raudvaramoodul kasutuses ainult Eesti Maaülikoolis ja plaanilises järjekorras TÜ Chemicumis. Viimases teevad tudengid juhtimissüsteemi kasutades oma lõputöid ja seadmed on 24/7 hõivatud. Ehk juhtimissüsteemi muutmine / modifitseerimine ei ole käesoleval hetkel mõistlik.

4.5.4 Virtualiseerimine

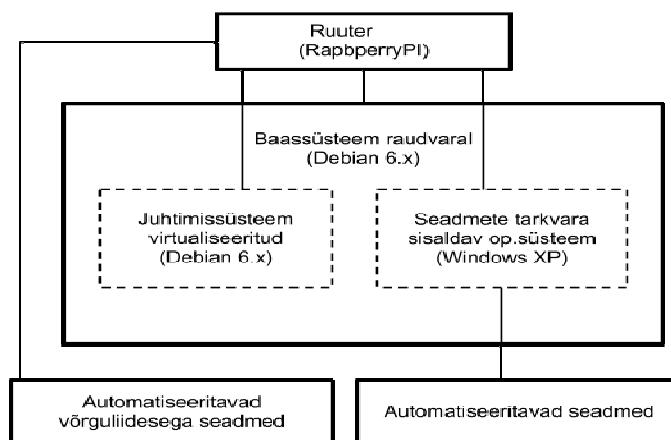
Juhtimissüsteem on soovitatav paigaldada virtualiseeritud keskkonda. Seda nii paremat raudvara utiliseerimist kui ka paigaldamiskiirusest silmas pidades. Samuti võimaldab eeltoodud konfiguratsioon kasutada juba olemasolevat raudvara.

TÜ Chemicumis olev süsteem koosneb, töö kirjutamise hetkel, virtualiseerimise poolelt vaadelduna Windowsi põhjast, virtualiseeritud Linuxi juhtimissüsteemist ja kahest rauaalkooksvast Windowsi masinast. Mille küljes on automatiseeritavad seadmed.



Joonis 2. Chemikumis kasutusel oleva raudvara arhidektuur.

Maaülikoolis olev süsteem koosneb, töö kirjutamise hetkel, virtualiseerimise poole pealt Linuxi põhjast virtualiseeritud Windowsist ja virtualiseeritud Linuxil jooksvast juhtimissüsteemist. Raual jooksva süsteemina on testitud Debian 6.x ja Sientific Linux 6.4 [SL]. Test oli läbi viidud, et veenduda, et ei teki probleeme erinevaid pakihaldussüsteeme kasutavate linuxite virtualiseerimistarkvaraga. Katse näitas, et võib olla vajalik juhtimissüsteemi paigaldamisel võrguseadete muutmine. Näiteks ei suutnud kasutatav virtualiseerimisplatvorm leppida windosi keskkonda paigaldamisel, et juhtimissüsteemil on võrguliidesed wlanX/ethX. Probleemi kõrvaldamiseks piisas võrguliidese keelamisest/lubamisest virtualiseermisplatvormi kasutajaliideses. Võib öelda, et juhtimissüsteemi saab paigaldada Windows 7, Debian 6.x ja SL 6.x platvormile.



Joonis 3. Eesti Maaülikoolis kasutusel oleva raudvara arhidektuur.

Eesmärgi täitmiseks kasutatava juhtimistarkvara tarkvara maht on piisavalt suur, et selle uue koopia kompileerimine, paigaldamine, seadistamine võtab kordades rohkem aega kui kui terve operatsioonisüsteemi paigaldamine. Selle tõttu on juhtimissüsteem on paigaldatud Linuxi distributsioonile „Debian 6.x“ ja pakendatud OVF [OVF] formaati, mis on paljudes operatsioonisüsteemides paigaldatav kasutades valitud virtualiseerimiskeskonda. On olemas vastavad tööriistad, millega saab erinevalt pakendatud kettaformaate konverteerida, kui tekib vajadus paigaldada käesoleva töö raames kättesaadavaks tehtud süsteemi mõnele teisele testimata virtualiseerimiskeskonnale.

Virtualiseerimistarkvarana on kasutatav/testitud Virtualbox [virtualbox] ja Proxmox [proxmox]. Viimane eeldab protsessoril kvm [kvm] tuge. Ainult qemu [qemu] kasutamisel jääb süsteem liiga aeglaseks, et olla mõistlikult kasutatav. Tulenevalt püstitatud eesmärgist, hoida süsteem lihtne, on kasutuses Virtualbox kuna Proxmox on rohkem serveri lahendus koos oma klastrite ja migreerimisvahenditega. Mis on ilmne liialdus käesolevas kontekstis.

Baassüsteemi alla laadimisel tuleb arvestada orienteeruvalt ~2GB salvestuspinna kuluga.

4.5.4.1 Miinimumnõuded virtualiseeritud masinale

Kettapind	8GB. Kasutusel on LVM [LVM(8)] mis on peale mitmekordset salvestusruumi nihutamist ennast väga õigustanud.
Mälu	350MB (400MB swap). Mõistlik on anda kätte natuke rohkem sest süsteemis Sikuli võib olude kokkulangevusel soovida enam tarbida.
Võrguliides	1tk
CPU	1tk
Ekraan	12MB mälu

Tasub meeles pidada, et ressursside jagamisel ei jääks baassüsteem nn. 'nälga'. Sellise olukorra tekitamisel tõmbab virtuaalmasina kokku kukkumine kaasa ka rauast masina.

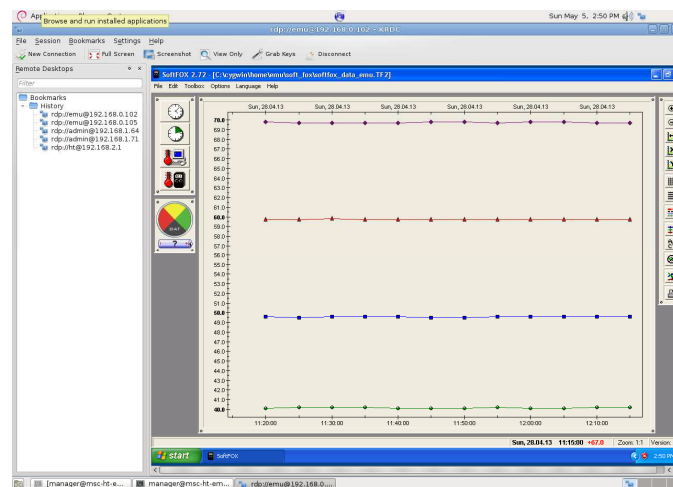
5 Näited

Järgnevalt on toodud mõned üksikud juhtimissüsteemi unikaalsemad rakendusnäited.

5.1 Takistumõõtja ”Scantronik Gigamodule”, andmelogeri ”Scantronik Termofox” ja ”RasperryPi” GPIO ümberlülitu liidestamine

Gigamodule koos oma andmelogoriga on mõeldud pikemaajalisteks mõõtmisteks. Vahepeal mõõtmistulemusi vaadata ei saa, kuna andmelogerial on tulemuste sisse- ja väljalugemiseks üks pistik (3 mm heli pistik). Ehk siis korraga saab ühendada kas arvuti või takistumõõtja. Sellest probleemist mööda minekuks tehakse mõõtmiste vaheaegadel RasperryPi GPIO'd kasutades ümberlülitus arvuti ja gigamodule vahel ja laetakse andmed juhtimissüsteemi arvutisse üle ssh.

Skripti `/usr/local/bin/pull_softfox.sh`. Kuna mõõdetud info salvestatakse algselt Windows XP masinasse, milles vaikimisi ei jookse ssh serverit, siis on paigaldatud Cygwin [cygwin] ja Opensshd [openssh] pakk. XP masinasse andmelogeri faili tekkivad unikaalsed read kopeeritakse juhtimissüsteemi arvutisse faili `/var/log/sec/misc/precisa.txt`. Kopeerimine on teostatud SEC'i [SEC] vixie croni [CRON] laadse funktsionaalsusega. Takistumõõtja ja andmelogeri kasutamine käib läbi alltoodud SoftFOX (vt pilti 2) nimelise kasutajaliidese. Viimane on automatiseeritud Sikuli `/usr/local/bin/sikuli/sbus/sbus.py` skripti abil. Kõiki skripte käivitatakse läbi alltoodud haldusliidese.



Pilt 2. Andme logeri ja takistumõõtja kasutajaliides Eesti Maaülikoolis.

5.2 Kaalu "Precisa" ja miniarvuti "RaspberryPi" liidestamine

Kaal väljastab mõõteandmeid seerialporti (rs232). Skript /usr/local/bin/precisa.sh mis töötleb kaalu väljundit, asub juhtimissüsteemi arvutis. Kaal ise on usb-rs232 konverteriga RaspberryPi küljes. Ühenduskanaliks kahe masina vahel on ssh tunnel. Skriptiga töödeldud väljund suunatakse juhtimissüsteemi arvutis olevasse faili /var/log/sec/misc/precisa.txt.

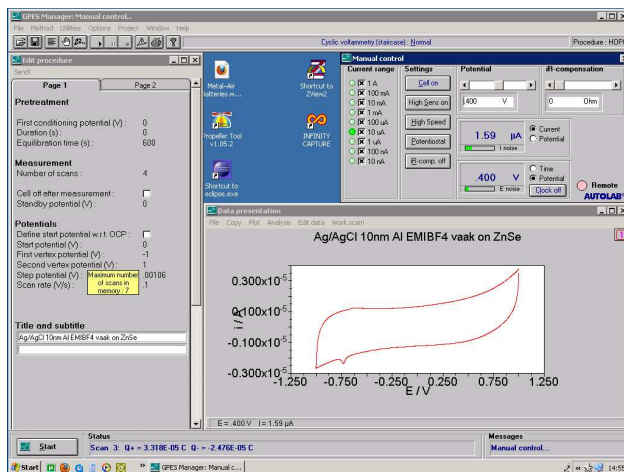
5.3 "AutoLab PGSTAT 30" kui elektroodide potentsiaali kontrolleri, "Spectrum GX" infrapuna spektromeeter'i liidestamine

Autolab ja Spectrum GX tarkvara (vt pilti 3) on paigaldatud kahte eraldiseisvasse Windows XP masinasse, sest ühes arvutis hakkasid need üksteist segama. Kuna tegemist on kahe erineva tootja poolt valmistatud aparaatidega, puudub neil ühendusprotokoll ehk võimalus üksteisega "suhelda".

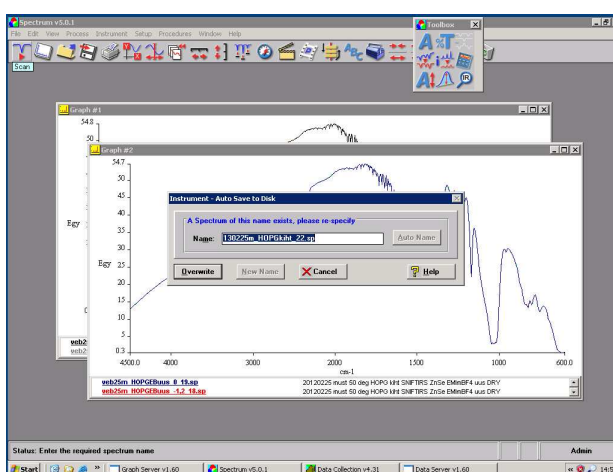
Tavo Romann (TÜ Chemicumi teadur): "Tegemist on küllaltki uudsete katsetega ja sellist aparaati, mis kontrolliks pinget ja mõõdaks infrapuna spektrit samaaegselt, keegi ei toodagi. Aparatuur on ka väga kallis, näiteks maksab Spectrum GX umbes 100 korda rohkem kui seda juhtiv arvuti. Töö eesmärgiks on analüüsida superkondensaatori pinget limiteerivaid reaktsioone ja tuvastada tekkinud ained.

Eelnev olukord: elektrokeemia (Autolab) ja spektrite (Spectrum GX) koos mõõtmiseks (tulemuseks siis spektroelektrokeemia) kirjutati Autolab'i tarkvara aknasse pinge väärtus (näiteks -1,6 V) ja Spektrum GX tarkvara aknasse sisestati faili nimeks pinge väärtus ja pandi arvuti spektrit mõõtma. Ühe katse käigus (kestvusega ~32 h) tuli seda protseduuri korrata sadu kordi. Väga ajakulukas tegevus ja pealegi oli lihtne parameetrite sisestamisel eksida."

Nendesse masinatesse tehakse juhtimisarvutist RDP'd (KRDC [krdc] tarkvara) kasutades korda mööda ühendusi (/usr/local/bin/sikuli/krdc/krdc.py) ja manipuleeritakse seadmete graafilist kasutajaliidest programmiga Sikuli (/usr/local/bin/sikuli/sbus/sbus.py). Ekraanipildilt loetakse pingete/voolude väärtuseid faili. /var/log/sec/sec/ui/ui.txt kasutades skripti /usr/local/bin/image2text.sh. Viimane kasutab peamiselt Tesseract [TESSERACT] mootorit ja ImageMagic [ImageMagic] pildi manipuleerimistööriistu. Automatiseeritavad seadmed on ligi pääsetavad läbi alltoodud kasutajaliidest. Graafilise kasutajaliidese automatiseerimise skriptide sisu on kirjutatud TÜ Chemikumi teadurite poolt.



AutoLab PGSTAT 30



Spectrum GX

Pilt 3. Chemikumis kasutatavad kasutajaliidesed.

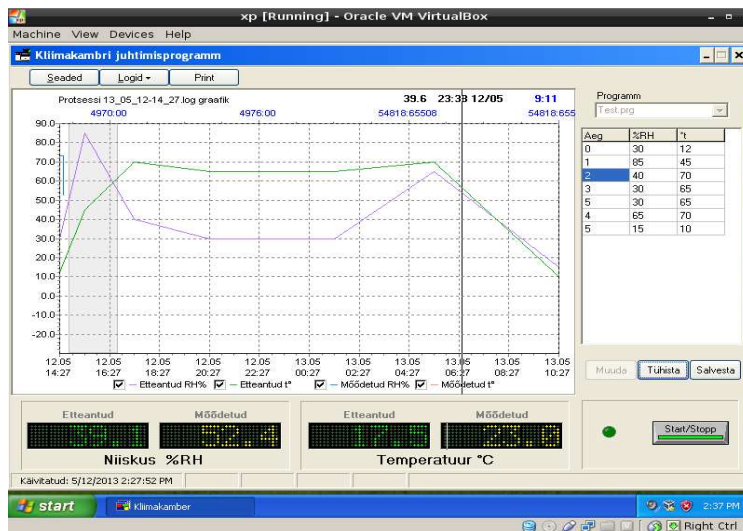
Tavo Romann: "Hannes Tamme töö tulemusena valmis tarkvaralahendus, mis klikib kordamööda Autolab'i ja Spectrum GX'i tarkvara akendes vastavatele nuppudele, sisestab ja loeb parameetreid ning käivitab infrapuna spektrite ja elektrokeemia mõõtmisi. "Tööjuhend" käesolevale tarkvarale on koostatud Sikuli abil, koosnedes paarikümnest reast koodist. Käesolev tarkvara teeb ära peaaegu terve inimese ööpäeva töö. Oleme valminud tarkvaraga mõõtnud ~50 katset (keskmise kestvusega ~32 h), mille baasil on ajakirja saadetud 1 artikkel (Romann et al. 2013) ja mõned on valmimisel. Tegemist on väga universaalse lahendusega ja tulevikus oleme huvitatud ühendamaks ka Raman spektroskoopi ja Autolab'i tarkvara."

5.4 Kliimakambri juhtimisprogrammis seadepunktide muutmine

Kliimakambri juhtimisprogrammis (vt pilti 4) sisestatakse juhtimisprogramm käsitsi tabeli kujul. Võimalik on muuta käesolevale tunnile järgnevate tundide soovitud seadepunkte. Eelnevalt tehti seda käsitsi. Seda tehakse selle pärast, et olemasolevad puidu kuivatamise mudelid, simulatsiooniprogrammid annavad häid tulemusi just lühikese perioodi jooksul.

Kliimakambri juhtimisprogrammi järgneva tunni andmete sisestamiseks antakse viimased parameetrina ette bash'i skriptile `/usr/local/bin/kliimakamber/kliimakamber.sh`. Viimane genereerib Sikuli poolt loetava faili (`/var/log/sec/misc/kliimakamber.conf`). Bashi skripti

käivitatakse läbi SEC'i poolt kuulatava faili . Viimane tekitab töö lõpus SEC'i poolt kuulatavas faili sündmuse mille peale "Kliimakambri juhtimisprogrammi" sisestatakse järgmise tunni seadepunktid skripti /usr/local/bin/sikuli/kliimakamber/kliimakamber.py poolt. Töö kirjutamise hetkel on veel ebaselge milliste parameetrite alusel soovitakse genereerida kliimakambris sisse laetava faili sisu. Ehk siis kuidas, milliste mudelitega simuleeritakse kuivatusprotsessi.



Pilt 4. Eesti Maaülikoolis kasutusel olev "Kliimakambri juhtimisprogramm" [Evikon].

6 Kokkuvõte

Eelnevalt kirjeldatud tarkvarakomplekt võimaldab ühendada erinevaid ühildumatuid süsteeme üheks automatiseeritavaks tervikuks. Uue seadme/tarkvara süsteemi lisamine ei eelda suuremahulist programmeerimistööd. Uue seadme lisamisel vajalik skripti ridade arv on kümnetes, mitte sadades ega tuhandetes, nagu see oleks uue tarkvara kirjutamisel.

Lahendus sobib hästi piiratud eelarve ja resursside keskkonda. Samuti ka kavandatakse suuremahulisse süsteemide ühendamise projekti, kui üks võimalik prototüüp. Mõeldav oleks süsteemi edasi arendada Eesti teadusuuringute tippkeskuste tarbeks. Lahendust ei ole soovitatav kasutada elutähtsate sõlmpunktide loomisel. Viimast eriti siis, kui tegemist on graafiliste kasutajliideste automatiseerimist nõudvate ülesannetega.

Käesoleva töö rakendusbaasiks on olnud Tartu Ülikoolis keemia instituudi labor ja Eesti Maaülikoolis MI metsatööstuse osakonna puiduteaduse labor.

Paigaldatud süsteemi kohaldamisel tundub, et töö autori osalust vaja ei ole. Seda võib väita Chemicumi teadurite tagasiside põhjal kellel kulus enda seadmete liidestamiseks süsteemiga umbes üks tööpäev. Süsteemi struktuuri muutmiseks võib olla vaja erialast kompetentsi omava inimese abi. Ehk siis käesoleva töö autori jaoks arusaamatu katseteseeria automatiseerimise saavad näidete põhjal oludele sobivaks kohandada süsteemi kasutajad, kuid süsteemi skeleti modifitseerimine jääb arvatavasti IT toe pärusmaaks.

Magistritöö üheks praktiliseks väljundiks on rutiinsete korduvate mõõteoperatsioonide automatiseerimise võimalus teadustegevuses, mis peaks vähendama peamiselt doktorantide töökoormust. Teiseks praktiliseks väljundiks on „raua“ ehk „arvutikastide“ arvu vähenemine.

Kokkuvõtlikult võib öelda, et eesmärk süsteemi lihtsana hoida on saavutatud osaliselt, kuid omavahel on võimalik liidestada, mida iganes, mida saab kirjutada failina.

7 Summary

Process management framework on the example of convective drying of wood and spectroelectrochemistry

The previously described software set allows connecting a variety of incompatible systems into a single automatic unit. Adding new equipment or a new software system does not require large-scale programming. Tens of lines for the script are required when adding a new device, instead of hundreds or thousands as would be the case in writing new software.

The solution is well suited for the limited budget and the environment RESOURCES, as well as for the planned large-scale project for connecting systems, as a possible prototype. The system could be further developed for use by Estonian centres of excellence in scientific research. The solution is not recommended for use in the creation of critical nodes, particularly when it comes to tasks demanding automation of graphic user interfaces. This research has been implemented in the Institute of Chemistry at the University of Tartu and in the laboratory of wood science at the Department of Forest Industry of the Estonian University of Life Sciences.

Application of the installed system implies that the author's participation is not required. This may be suggested based on feedback from researchers working at Chemicum, the University of Tartu, who interfaced their own devices with the system within one workday. The help of a person with professional competence may be needed to change the structure of the system. Perhaps system users can adapt the automation test series, which is confusing for the author of this research, to the conditions although the system skeleton modification will probably be done by the IT support.

One of the practical outputs of this Master's thesis is the possibility of automation of routine repeated measuring operations in research, which should reduce the workload of doctoral students primarily. Another practical output involves a decrease in the number of hard drives, that is, computer cases.

In conclusion, it may be stated that the purpose of keeping the system simple was partially achieved, but interfacing can be done with anything that can be written to a file.

8 Viited

[DEBIAN]	http://www.debian.org (12-04-2013)
[SL]	https://www.scientificlinux.org (12-04-2013)
[MAN(1)]	man - an interface to the on-line reference manuals
[BASH(1)]	bash - GNU Bourne-Again Shell
[GPL]	http://www.gnu.org/licenses/gpl-2.0.html (12-04-2013)
[SUDO(8)]	sudo, sudoedit - execute a command as another user
[SUDOERS(5)]	sudoers - list of which users may execute what
[UMASK(2)]	umask - set file mode creation mask
[RV]	http://ristov.users.sourceforge.net (12-04-2013)
[SEC(1)]	sec - simple event correlator
[DAEMON(3)]	daemon - run in the background
[PERL(1)]	practical Extraction and Report Language
[ts(7)]	quantification various properties of objects
[SIKULI(1)]	sikuli - run sikuli scripts
[EXPECT(1)]	expect - programmed dialogue with interactive programs, Version 5
[PYFLAKES(1)]	pyflakes - simple Python source checker
[TESSERACT(1)]	tesseract - command-line OCR engine
[SSH(1)]	ssh - OpenSSH SSH client (remote login program)
[ImageMagick(1)]	ImageMagick - is a free software suite for the creation, modification and display of bitmap images.
[krdc]	http://docs.kde.org/stable/en/kdenetwork/krdc/index.html (12-04-2013)
[SourceForge]	http://sourceforge.net (12-04-2013)
[FIFO(7)]	fifo - first-in first-out special file, named pipe
[OVF]	http://en.wikipedia.org/wiki/Open_Virtualization_Format (12-04-2013)
[OCR]	http://en.wikipedia.org/wiki/Optical_character_recognition (12-04-2013)
[XAUTH(1)]	xauth - X authority file utilit
[R(1)]	R - a language for data analysis and graphics
[RFC 4180]	Common Format and MIME Type for Comma-Separated Values (CSV) Files

[parallel]	http://www.gnu.org/software/parallel (12-04-2013)
[SCREEN(1)]	screen - screen manager with VT100/ANSI terminal emulation
[TAIL(1)]	tail - output the last part of files
[rlwrap(1)]	rlwrap - readline wrapper
[HG(1)]	hg - Mercurial source code management system
[CM(8)]	cm - central Configuration Manager
[GREP(1)]	grep, egrep, fgrep, rgrep - print lines matching a pattern
[SED(1)]	sed - stream editor for filtering and transforming text
[MAWK(1)]	mawk - pattern scanning and text processing language
[TR(1)]	tr - translate or delete characters
[CUT(1)]	cut - remove sections from each line of files
[SSH-ADD(1)]	ssh-add — adds RSA or DSA identities to the authentication agent
[IPTABLES(8)]	iptables — administration tool for IPv4 packet filtering and NAT
[RasperryPi]	http://www.raspberrypi.org (12-04-2013)
[cygwin]	http://www.cygwin.com (12-04-2013)
[openssh]	http://www.openssh.org (12-04-2013)
[Evikon]	http://www.evikon.ee (12-04-2013)
[p8x32a]	http://www.parallaxsemiconductor.com/sites/default/files/parallax/Propeller-P8X32A-Datasheet-v1.4.0_1.pdf
[GPIO]	http://en.wikipedia.org/wiki/General_Purpose_Input/Output (12-04-2013)
[virtualbox]	https://www.virtualbox.org/wiki/Documentation (12-04-2013)
[proxmox]	http://pve.proxmox.com/wiki/Main_Page (12-04-2013)
[LVM(8)]	lvm - LVM2 tools
[kvm]	http://www.linux-kvm.org/page/Main_Page (12-04-2013)
[qemu]	http://wiki.qemu.org/Main_Page (12-04-2013)

T. Romann, O. Oll, P. Pikma, H. Tamme, E. Lust, Surface chemistry of carbon electrodes in EMImBF₄ ionic liquid – an in situ infrared study. Carbon, submitted (ajakirja saadetud).

Tamme, V., Muiste, P., Polyachenko, R., Tamme, H. (2010) Determination of Dynamics of Moisture content, Temperature and Mechanical Stress of Pine Wood During Convective Drying. In: 11th International IUFRO Wood Drying Conference, Skellefteå, Sweden, 2010. **Proceedings:** International IUFRO Wood Drying Conference, 2010, p. 147 – 154.

Tamme, V. ; Muiste, P.; Mitt, R.; Tamme, H. Determination of Effective Diffusion Coefficient and Mechanical Stress of Pine Wood During Convective Drying. **Baltic Forestry**, v. 17, p. 110 – 118, (2011).

Tamme, V., Muiste, P., Kask, R., Padari, A., Tamme, H. (2012). The research of electrode effects on wood drying above fiber saturation point. - 12th International IUFRO Wood Drying Conference „Challenges and Opportunities Related to Tropical Lumber Drying“. July 30 to August 03, 2012. Belém, Para, Brazil.

Tamme, V., Kask, R., Muiste, P., Tamme, H.(2012). Puidu niiskusmõõtjate võrdlus laiendatud mõõtepiirkonnas. – Neljateistkümnenda konverentsi ”Taastuvate energiaallikate uurimine ja kasutamine” kogumikus, lk.98 -108, 8. novembril, 2012. Tartu, Estonia.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Hannes Tamme (25.11.1982)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
”Protsessi juhtimise tarkvaraline raamistik puidu konvektiivkuivatuse ja spektroeletrokeemia näidetel” mille juhendajd on Madis Noppel ja Valdek Tamme.
 - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace’i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 23.05.2013